

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

SUPPLEMENTAL APPEAL BRIEF - 37 C.F.R. 1.192

U.S. Patent Application 09/890,536 entitled,
“METHOD AND SYSTEM FOR DYNAMIC CONFIGURATION OF ACTIVATORS
IN A CLIENT-SERVER ENVIRONMENT”

Real Party in Interest: IONA Technologies, Inc.

Related Appeals and Interferences:

An Appeal Brief was previously filed on 4/30/2007, which resulted in the Examiner withdrawing the Final Rejection and issuing a new Non-Final Rejection dated 08/08/2007. Applicants, via the current Supplemental Appeal Brief, request the Board to reinstate the previous Appeal as the Examiner still maintains the art of record fails to teach or suggest various features of the pending claims.

Status of Claims:

Claims 1-10 are pending.

Claims 1, 3-6, and 8-10 are rejected under 35 U.S.C. § 103(a) as being unpatentable over “The Common Object Request Broker: Architecture and Specification” (Hereafter, CORBA) in view of Cavanaugh (US 6,516,354).

Claims 2 and 7 are rejected under 35 U.S.C. §103(a) as being unpatentable over CORBA in view of Cavanaugh, and further in view of Chandy (US 6,898,791).

Claims 1-10 are hereby appealed.

Status of Amendments:

No after-final amendments were filed after the Final Office Action dated 12/29/2006, and the Non-Final Office Action dated 08/08/2007.

SUMMARY OF THE CLAIMED SUBJECT MATTER:

(NOTE: All references made in this section pertain to related **PCT/US00/02014** filed January 28, 2000)

According to claim 1, the present invention provides for a computer implemented method of activating a process, said method implemented via computer program code stored in said computer, said method comprising: generating one or more first plug-ins each configured to activate a target process (**see abstract, page 5, lines 1-5, lines 16-19 of PCT/US00/02014**); dynamically registering the first plug-ins with a second plug-in (**see abstract, page 5, lines 1-5, lines 18-19, page 8, lines 21-22, page 16, lines 25-27, and page 19, lines 18-19 of PCT/US00/02014**); permanently storing information relating to each registered first plug-in, and activating at least one target process based on said permanently stored information (**see abstract, page 5, lines 1-5, lines 16-19, page 8, lines 21-22, and page 16, lines 28-29 of PCT/US00/02014**).

In addition to the features of claim 1, claim 2 further comprises the steps of: storing a flag for each registered first plug-in; perpetually activating the corresponding target process if the flag is set to a first state; and activating the corresponding target process upon a request if the flag is set to a second state (**see page 5, lines 6-9 and lines 20-23 of PCT/US00/02014**).

In addition to the features of claim 2, claim 3, further comprises the steps of: generating an exception to indicate that a target process is inactive when its flag is not set to the first state or the second state (**see page 5, lines 10-13 and lines 24-26 of PCT/US00/02014**).

In addition to the features of claim 1, claim 4 further comprises the steps of: providing a unique identifier for each target process; and sending and receiving a message between the first and second plug-ins using the identifiers (**see page 5, lines 10-13 and lines 27-29 of PCT/US00/02014**).

In addition to the features of claim 4, claim 5 further comprises a method wherein the message includes information relating to a state change of the target processes, and wherein the state includes an activated state and a deactivated state (**see page 5, lines 29-30 of PCT/US00/02014**).

According to claim 6, the present invention provides a server computer in a client-server computer system, comprising: a processing unit; and a storage device storing computer program code implementing at least: one or more first plug-ins each configured to activate a target process (**see abstract, page 5, lines 1-5, lines 16-19 of PCT/US00/02014**), and a second plug-in configured to dynamically register the first plug-ins (**see abstract, page 5, lines 1-5, lines 18-19, page 8, lines 21-22, page 16, lines 25-27, and page 19, lines 18-19 of PCT/US00/02014**) and to permanently store information relating to the registered first plug-ins (**see abstract, page 5, lines 1-5, lines 16-19, page 8, lines 21-22, and page 16, lines 28-29 of PCT/US00/02014**).

In addition to the features of claim 6, claim 7 teaches a server computer wherein the second plug-in comprises: a memory configured to store a flag for each registered first plug-in, wherein the second plug-in is further configured to perpetually activate target processes having

their flags set at a first state and to activate target processes, upon receiving a request, having their flags set at a second state **(see page 5, lines 6-9 and lines 20-23 of PCT/US00/02014).**

In addition to the features of claim 7, claim 8 teaches a server computer wherein the second plug-in is further configured to generate an exception to indicate that the target process is inactive when the flag is not set to the first state or the second state **(see page 5, lines 10-11 and 24-26 of PCT/US00/02014).**

In addition to the features of claim 6, claim 9 teaches a server computer further comprising: a first computer program object configured to provide a unique identifier for each target process and configured to send a message using the identifiers **(see page 5, lines 10-13 and lines 27-29 of PCT/US00/02014).**

In addition to the features of claim 9, claim 10 teaches a server computer wherein the message includes information relating a state change of the target processes, and wherein the state includes an activated state and a deactivated state **(see page 5, lines 29-30 of PCT/US00/02014).**

GROUND OF REJECTION TO BE REVIEWED ON APPEAL:

Claims 1-10 are hereby appealed.

• **ISSUES:**

1. Was an ambiguous rejection provided under 35 U.S.C. §103(a) with respect to claims 3 and 8?

2. Claims 1-10 are pending. Claims 1, 3-6, and 8-10 are rejected under 35 U.S.C. § 103(a) as being unpatentable over “The Common Object Request Broker: Architecture and Specification” (Hereafter, CORBA) in view of Cavanaugh (US 6,516,354). Claims 2 and 7 are rejected under 35 U.S.C. §103(a) as being unpatentable over CORBA in view of Cavanaugh, and further in view of Chandy (US 6,898,791). Was an improper 35 U.S.C. §103(a) rejection issued with respect to claims 1-10?

• **ARGUMENT:**

1. Was an ambiguous rejection provided under 35 U.S.C. §103(a) with respect to claims 3 and 8?

In response to the Appeal Brief of 04/30/2007, the Examiner has issued a new Non-Final Action, wherein the Examiner has rejected dependent claims 3 and 8 under 35 U.S.C. § 103(a) as being unpatentable over “The Common Object Request Broker: Architecture and Specification” (Hereafter, CORBA) in view of Cavanaugh (US 6,516,354). However, it should be noted that dependent claim 3 depends on dependent claim 2 and dependent claim 8 depends on dependent claim 7. It should also be noted that in the new Non-Final Office Action of 08/08/2007, the Examiner has rejected dependent claims 2 and 7 under 35 U.S.C. §103(a) as

being unpatentable over **CORBA** in view of **Cavanaugh**, and further in view of **Chandy** (US 6,898,791). Applicants are unsure how the Examiner is rejecting claim 2 using three references (i.e., CORBA, Cavanaugh, and Chandy), while rejecting claim 3 that depends on claim 2 using only two references (i.e., CORBA and Cavanaugh). Applicants are also unsure how the Examiner is rejecting claim 7 using three references (i.e., CORBA, Cavanaugh, and Chandy), while rejecting claim 8 that depends on claim 7 using only two references (i.e., CORBA and Cavanaugh). Hence, at least for this reason, **Applicants respectfully assert that an improper and ambiguous rejection was issued with regards to dependent claims 3 and 8.**

2. **Was an improper 35 U.S.C. §103(a) rejection issued with respect to claims 1-10?**

Claims 1, 3-6, and 8-10 are rejected under 35 U.S.C. § 103(a) as being unpatentable over “**The Common Object Request Broker: Architecture and Specification**” (Hereafter, CORBA) in view of Cavanaugh (US 6,516,354). Claims 2 and 7 are rejected under 35 U.S.C. §103(a) as being unpatentable over CORBA in view of Cavanaugh, and further in view of Chandy (US 6,898,791).

The CORBA reference outlines the CORBA standard and details how it defines APIs, communication protocol, and object/service information models to enable heterogeneous applications written in various languages running on various platforms to interoperate.

Applicants’ independent claim 1, by contrast, provides for a computer implemented method of activating a process, said method implemented via computer program code stored in

said computer, said method comprising: generating one or more first plug-ins each configured to activate a target process; dynamically registering the first plug-ins with a second plug-in; permanently storing information relating to each registered first plug-in, and activating at least one target process based on said permanently stored information.

With respect to the rejection of independent claim 1, on page 2 of the new Non-Final Office Action of 08/08/2007, the Examiner states that Applicants' feature of "generating one or more first plug-ins" can be equated to CORBA's created Portable Object Adapters (POAs) as described in §9.2.3. Further, with respect to Applicants' feature of each plug-in being configured to activate a target process, the Examiner adds the new clarification that §9.3.3, "especially p.949", teaches such a feature. Applicants wish to note that the newly added reference to page 949 references a section in the INDEX of the CORBA document. Applicants are unsure what the Examiner is referencing in the INDEX as providing support for the feature of each plug-in being configured to activate a target process. At least with respect to this new citation, Applicants respectfully assert that the Examiner has issued an ambiguous rejection.

On page 3 of the new Non-Final Office Action of 08/08/2007, the Examiner cites §9.3.2, §9.3.3, §1.3.2.1, and Figure 11-3 of CORBA as teaching Applicants' feature of activating at least one target feature. As was pointed out in the Appeal Brief of 04/30/2007, the Examiner's citations of §9.3.2 and §9.3.3 deal with different entities, i.e., POA Manager Interface and Adapter Activator Interface, respectively. Applicants are unsure how the Examiner has used the citations with respect to the POA Manager Interface and the Adapter Activation Interface with regards to the argument that these interfaces provide the feature of activating at least one target feature.

Also, as was pointed out in the Appeal Brief of 04/30/2007, it appears that the Examiner is taking a mere mention of the term “**activate**” in §9.3.2 and §9.3.3 and has erroneously equated such a term to the Applicants’ feature of “activating at least one target process”. Specifically, the Board is respectfully requested to review §9.3.2 and §9.3.3 which merely teach various “**Processing States**” which, by CORBA’s own admission, outlines the **processing states associated with a POA Manager**. See also, for example, CORBA’s own statement on page 9-15 (section entitled “Processing States”), which states that “A **POA Manager** has four possible processing states: active, inactive, holding, and discarding” (emphasis added).

The Examiner has erroneously concluded that §9.3.2 teaches POAs that are configured to activate a target process based on the mere mention of the phrase “active state”. However, the Board is respectfully requested to review figure 9-3 (reproduced below) and its associated description, which explicitly states that: (1) these states are associated with the POA Manager, and (2) the “active state” reference used by the Examiner does NOT refer to activating a target process as the Examiner asserts, but merely refers to **transitioning the POA Manager from a “discarding state” to an “active state”**.

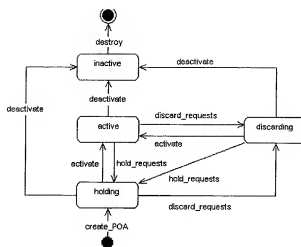


FIGURE 9-3 of CORBA

Applicants also wish to emphasize that the Examiner’s citation of **§1.3.2.1** is **not found anywhere in the CORBA reference** as the last entry for Chapter 1 (“**The Object Model**”) ends at §1.3.2 entitled “The Construction Model”. At least with respect to the citation of **§1.3.2.1**, Applicants respectfully assert that the Examiner has issued an ambiguous rejection.

With respect to the Examiner’s citation of **Figure 11-3** (which is reproduced below), Applicants respectfully assert that Figure 11-3 merely teaches the **configuration of Bridges in networks** and is completely irrelevant to Applicants’ feature of activating a target process. At least with respect to the citation of Figure 11-3, Applicants respectfully assert that the Examiner has issued an ambiguous rejection.

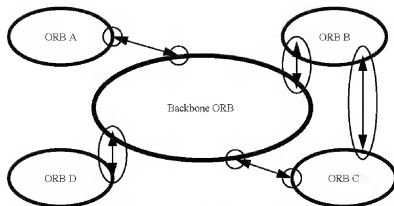


Figure 11-3 An ORB chosen as a backbone will connect other ORBs through bridges, both full-bridges and half-bridges.

Further, with respect to claim 1, the Examiner asserts that Applicants' feature of "dynamically registering the first plug-in with a second plug-in" is taught by CORBA's statement regarding "an AdapterActivator object must...registered with". It should be emphasized that the **Examiner has equated the POA to Applicants' first plug-in** and the **Examiner has equated the AdapterActivator Object to Applicants' second plug-in.** However, **the Examiner has failed to indicate where in the CORBA reference is a teaching or suggestion for the POA to be dynamically registered with the AdapterActivator Object.** The Examiner's citation, which appears in §9.3.3, merely mentions that "an AdapterActivator object must be local to the process containing the POA objects it is registered with" (emphasis added). This merely mentions that the **AdapterActivator object is registered with a process containing the POA objects.** However, there is no teaching or suggestion for a **first plug-in to dynamically register with a second plug-in.**

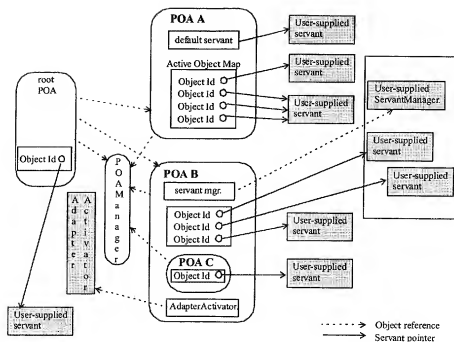


FIGURE 9-2 of CORBA

In the new rejection of 08/08/2007, the Examiner correctly concludes that the CORBA reference fails to teach the feature of activation being based on permanently stored information. However, Applicants respectfully disagree with the Examiner's conclusion that such as feature is remedied by Cavanaugh.

Cavanaugh teaches a distributed object systems that uses the CORBA architecture and, more particularly, teaches group object identification in object adapters.

For support of his/her argument, the Examiner cites columns 7 and 8 of Cavanaugh as teaching such permanent storage. However, by Cavanaugh's own admission, what is stored is **"POA Name to POA ID mapping"** (see column 7, lines 59-64 of Cavanaugh). Cavanaugh

further gives examples of mappings such as “Root/A/D -> 10” and “Root/B/D->11”. Given Cavanaugh’s own words that the mapping between POA Name and POA ID is what is stored, it would be erroneous to argue that Cavanaugh teaches or suggests permanently storing information relating to each registered plug-in. Similarly, it would also be erroneous to argue that Cavanaugh teaches activating a target process based on the permanently stored information.

Applicants, therefore, respectfully assert that the combination of CORBA and Cavanaugh cannot anticipate or render obvious Applicants’ claim 1.

Applicants’ independent claim 6, provides for a server in a client-server computer system, comprising: a processing unit; and a storage device storing computer program code implementing at least: one or more first plug-ins each configured to activate a target process, and a second plug-in configured to dynamically register the first plug-ins and to permanently store information relating to the registered first plug-ins.

The above-mentioned arguments substantially apply to independent claim 6.

At least for the reasons set forth above, Applicants respectfully assert that the combination of CORBA and Cavanaugh cannot anticipate nor render obvious Applicants’ independent claim 6.

With respect to claims 4-5 and 9-10, the Examiner equates the mere mention of the “delivery of requests to the POA” (see §9.3.3 of CORBA) to Applicants’ feature of sending a message using the identifiers. §9.3.3 of CORBA merely mentions the **delivery of a request during initialization of a POA** and provides no teaching or suggestion for **sending messages using identifiers**.

Also, the above-mentioned arguments with respect to independent claims 1 and 6 substantially apply to claims 3-4 and 8-10 as they inherit all the features of the claim from they depend from (i.e., claims 1 and 6, respectively).

Hence, the combination of CORBA and Cavanaugh cannot anticipate nor render obvious Applicants’ dependent claims 3-4 and 8-10.

Claims 2 and 7 are rejected under 35 U.S.C. §103(a) as being unpatentable over CORBA in view of Cavanaugh, and further in view of Chandy (US 6,898,791). Applicants wish to note that the Office Action of 08/08/2007 erroneously cited the Chandy reference as **US 6,96,791**. However, Applicants’ representative clarified this rejection with the Examiner and the Examiner clarified that the number cited was in error and the correct number for the Chandy reference is **US 6,898,791**. It should be noted that the argument presented below is with reference to US Patent 6,898,791.

Chandy teaches a distributed system framework for a networked environment, including objects such as: a program method for creating at least one inbox for storing messages received

from another process object; a program method for creating at least one outbox for storing messages to be transmitted to another process object; a **freeze method that saves a state of the process object to persistent storage**; a thaw method that restores the frozen process object from the persistent storage; or a program method for interconnecting each created outbox of the process object to a created inbox of at least one other process object.

Chandy teaches in column 8, lines 21-38 that the freezing method merely involves freezing a process and makes no mention, as the Examiner asserts, of a flag that is maintained in persistent storage for each registered plug-in.

Absent such a teaching, Applicants respectfully assert that the combination of CORBA, Cavanaugh, and Chandy cannot anticipate nor render obvious Applicants' claims 2 and 7.

As many of the features of pending claims 1-10 are not taught by the combination of CORBA, Cavanaugh, and Chandy, Applicants respectfully assert that an improper 35 U.S.C. §103(a) rejection was given by the Examiner. Applicants respectfully request the board to reverse the Examiner's rejection with respect to claims 1-10.

SUMMARY

Applicant contends that the Examiner, in the office action of 12/29/2006, has failed to provide a *prima facie* case of obviousness under U.S.C. §103 as there is no suggestion or motivation, either in the cited references, or in the knowledge generally available to one of ordinary skill in the art, to modify the references to provide for the various features of Applicants' invention. As has been detailed above, none of the references, cited or applied, provide for the specific claimed details of Applicants' presently claimed invention, nor render them obvious. It is believed that this case is in condition for allowance and reconsideration thereof, and early issuance is respectfully requested.

This Supplemental Appeal Brief is being filed with a fee for an extension of time. The Commissioner is hereby authorized to charge any deficiencies in the fees provided to Deposit Account No. 50-4098.

Respectfully submitted by
Applicant's Representative,

/ramraj soundararajan/

Ramraj Soundararajan
Reg. No. 53,832

IP Authority, LLC
9435 Lorton Market Street #801
Lorton, VA 22079
(571) 642-0033

December 10, 2007

CLAIMS APPENDIX:

1. (Previously Presented) A computer implemented method of activating a process, said method implemented via computer program code stored in said computer, said method comprising:

generating one or more first plug-ins each configured to activate a target process;
dynamically registering the first plug-ins with a second plug-in;
permanently storing information relating to each registered first plug-in, and
activating at least one target process based on said permanently stored information.

2. (Original) The method of claim 1 further comprising:

storing a flag for each registered first plug-in;
perpetually activating the corresponding target process if the flag is set to a first state; and
activating the corresponding target process upon a request if the flag is set to a second state.

3. (Original) The method of claim 2 further comprising:

generating an exception to indicate that a target process is inactive when its flag is not set to the first state or the second state.

4. (Original) The method of claim 1 further comprising:

providing a unique identifier for each target process; and

sending and receiving a message between the first and second plug-ins using the identifiers.

5. (Original) The method of claim 4 wherein the message includes information relating to a state change of the target processes, and wherein the state includes an activated state and a deactivated state.

6. (Previously Presented) A server computer in a client-server computer system, comprising:

a processing unit; and

a storage device storing computer program code implementing at least:

one or more first plug-ins each configured to activate a target process, and

a second plug-in configured to dynamically register the first plug-ins and to permanently store information relating to the registered first plug-ins.

7. (Original) The server of claim 6 wherein the second plug-in comprises:

a memory configured to store a flag for each registered first plug-in, wherein the second plug-in is further configured to perpetually activate target processes having their flags set at a first state and to activate target processes, upon receiving a request, having their flags set at a second state.

8. (Original) The server of claim 7 wherein the second plug-in is further configured to generate an exception to indicate that the target process is inactive when the flag is not set to the first state or the second state.

9. (Original) The server of claim 6 further comprising:

a first computer program object configured to provide a unique identifier for each target process and configured to send a message using the identifiers.

10. (Original) The server of claim 9 wherein the message includes information relating a state change of the target processes, and wherein the state includes an activated state and a deactivated state.

EVIDENCE APPENDIX:

None

RELATED PROCEEDINGS APPENDIX:

None